SLIIT

Discover Your Future

Topic : Online Grocery Delivery

Group no : MLB_PG.05.02_08

Campus : Malabe

Submission Date : 2020/10/20

We declare that this is our own work and this Assignment does not incorporate without acknowledgment any material previously submitted by anyone else in SLIIT or any other university/Institute. And we declare that each one of us equally contributed to the completion of this Assignment.

Registration No	**Name**	**Contact Number**

1. Customer should register providing details such as name, e-mail address, contact number to buy items.
2. The customer can view the grocery items and add the desired items to the shopping cart.
3. The customer can check the shopping cart and add or remove items from the cart and place the order.
4. The customer can choose a payment method such as card payment or cash on delivery to purchase the item.
5. The bank assistant validate the card details (name of the card holder, card type, expiry date, cvc code) of the customer.
6. The customer has to fill a form providing his/her home address, delivery time slot and date after placing the order.
7. Admin can verify all the order of the online grocery delivery.
8. Customer receives a confirmation e-mail after verifying the order.
9. The delivery person should take addresses and contact numbers of customer list and then track the location.
10. The automated bill of each customer is issued to the delivery person.
11. He/she should deliver the grocery items along with the bill to the customer within the respective time.
12. If the payment method is cash on delivery, delivery person should collect the cash and record the amount of cash paid.
13. Admin can add new items and update the exist items in the store, restock or delete items from the store, generate a list of items that need to be restock.
14. The marketing advertiser can set up promotion details, maintain user interface.

<u>Noun- Verb Analysis</u>

Customer	Class
Name	Attribute
E-mail address	Attribute
Contact number	Attribute
Grocery items	Class
Form	Attribute
Shopping cart	Class
Payment	Class
Card payment	Redundant
Cash on delivery	Redundant
Bank assistant	Outside scope
Home address	Attribute
Delivery time slot	Attribute
Date	Attribute
Order	Class
Admin	Outside scope
Delivery	Class
Delivery person	Outside scope
Location	Attribute
Bill	Class
Respective time	Attribute
Marketing Advertiser	Outside scope
Store	Outside scope

Customer – Register, “Check e-mail”, Store details of customer.

Item – Restock, Remove, Update

Shopping cart – Add , Remove, Check, Store details of items in the shopping cart.

Payment – Validate, “Collect cash”

Order – “Place order”, “Verify the order”, ”status of order”.

Delivery – Deliver, Track location, “Record the amount of cash paid”

Report – “List of items that need to be restock”, “List of details of customers”, “List of previous orders”, ”List of items in the shopping cart”.

Bill – “Issue the bill”, “Details of ordered items.”, Store details of items.

Classes

- Customer
- Grocery items
- Shopping cart
- Payment
- Order
- Delivery
- Report
- Bill

Class Name: Customer	
Responsibilities:	**Collaborations:**
Register entering details.	
Store details of customer.	
Check emails.	

Class Name: Shopping Cart	
Responsibilities:	**Collaborations:**
Store details of items in the shopping cart.	
Add items.	Item.
Remove items.	Item.
Check items in the shopping cart.	Item.

Class Name: Item	
Responsibilities:	**Collaborations:**
Store details of items.	
Update items.	
Restock.	

Class Name: Order	
Responsibilities:	**Collaborations:**
Place order.	Item.
Status of orders.	Item
Store details of ordered items.	Item, customer.
Verify.	

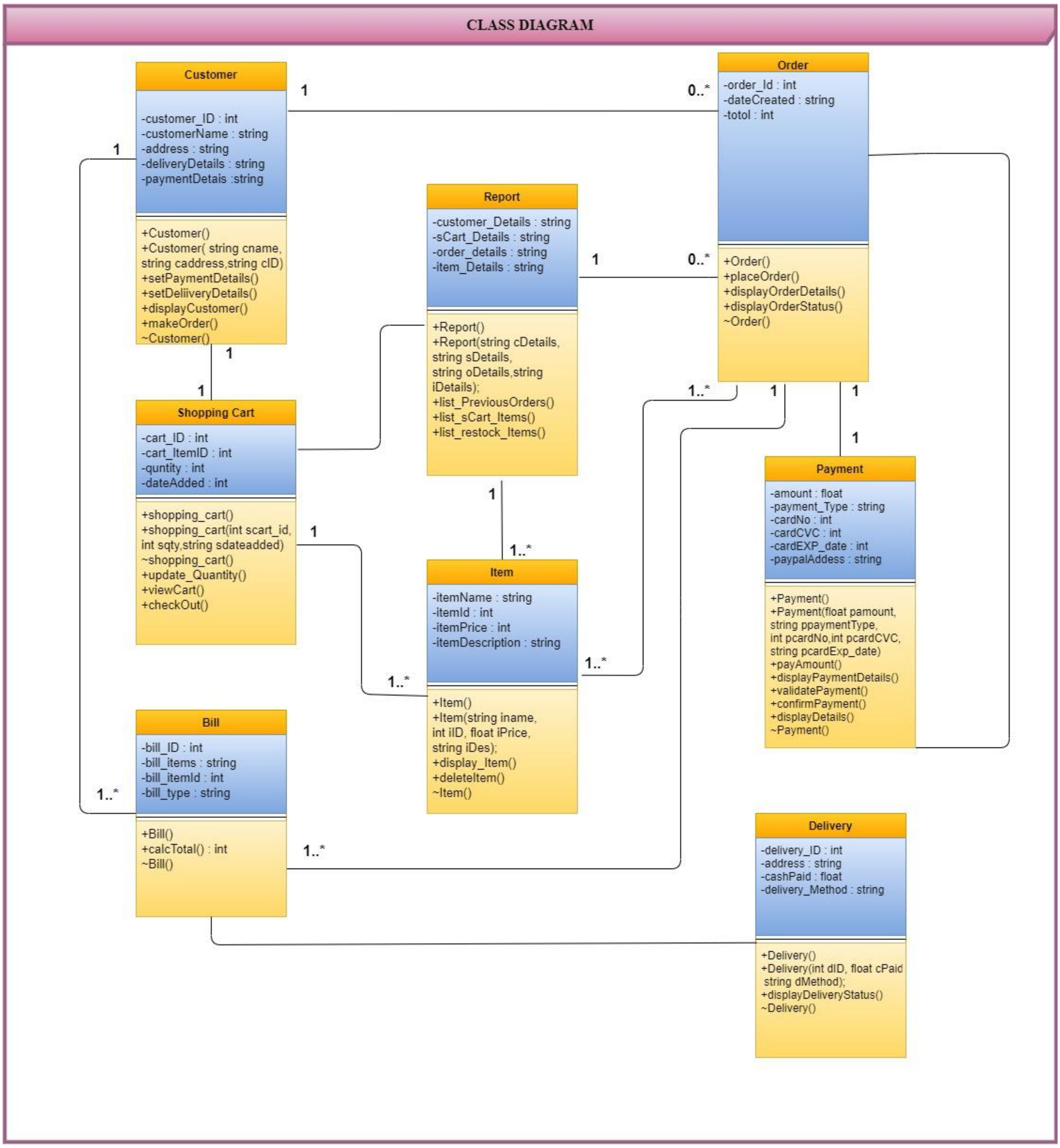
Class Name: Delivery	
Responsibilities:	**Collaborations:**
Deliver the orders.	Order.
Track location.	
Record the amount of cash paid.	

Class Name: Payment	
Responsibilities:	**Collaborations:**
Store payment details.	
Validate.	
Collect cash.	

Class Name: Bill	
Responsibilities:	**Collaborations:**
Details of ordered items.	Order, Customer.
Issue the bill.	

Class Name: Report	
Responsibilities:	**Collaborations:**
List of previous orders.	Customer, Order.
List of items in the shopping cart.	Item.
List of restock items.	Item.
List of details of customers.	Customer.

Class Diagram



```cpp
#include <iostream>
#include <string>
using namespace std;

class Customer;
class Item;
class  shopping_cart;
class Payment;
class Order;
class Delivery;
class Bill;
class Report;

//customer class

class Customer
{
  private:
    string customerName;
    string address;
    string customer_ID;
    string deliveryDetails;
    string paymentDetais;
    shopping_cart * shoppingc;
    Order * order[5];
    Bill * bill[6];


  public:
    Customer();
    Customer( string cname, string caddress, string cID);
    void assignShoppingCart(shopping_cart * sc);
    void setDeliveryDetails();
    void MakeOrder(int i,Order * ord);
    void assignBill(int x,Bill * bl);
    void setPaymentDetails();
    void displayCustomer();
    ~Customer();

};


//Order class


class Order
{
   private:
    Customer *Cus;
    string dateCreated;
    string orderID;
    int total;
    Customer * cus;
    Item * tyu[9];
    Payment * p;

   public:
    Order();
    Order (string corderID,  string date,Customer * c,Payment * p1);
    ~Order();
    void orderItem(int v,Item * tyul);
    void displayOrderDetails();
    void displayOrderStatus();
    void displayOrders();
};


//Item class


class Item
{
private:
  string itemName;
  string itemID;
  float itemPrice;
  string itemDescription;
  shopping_cart * qwe;
  Report *repa;
  Order *orc[8];

public:
  Item();
  ~Item();
  Item(string iname, string iID, float iPrice, string iDes,shopping_cart *
qw,Report *repal);
  void assignOrderItem(int r,Order *orct);
  void displayItem();
  void deleteItem();
};


//shopping_cart class


class shopping_cart
{
  private:
      string cart_id;
      int qty;
      string dateadded;
      Customer * cus;
      Item * itm1[10];
      Report * sml;

  public:
      shopping_cart();
      shopping_cart(string scart_id,int sqty,string sdateadded,Customer *
cust,Report * sml1);
      void addCartItem(int y,Item * it);
      void update_Quantity();
      void viewCart();
      void checkout();
      ~shopping_cart();

};


//Payment class


class Payment {
    private:
        float amount;
        string paymentType;
        int cardNo;
        int cardCVC;
        string cardExp_Date;
        Order *oq;

    public:
        Payment();
        ~Payment();
        Payment(float pamount,string ppaymentType,int pcardNo,int pcardCVC,
string pcardExp_date,Order *oq1 );
        void validatePayment();
        void confirmPayment();
        void displayDetails();
  };



//Delivery class


class Delivery
{
  private:
    string delivery_ID;
    float cashPaid;
    string delivery_Method;

  public:
  Delivery();
  ~Delivery();
  Delivery(string dID, float cPaid, string dMethod);
  void displayDeliveryStatus();
};


//Bill class


class Bill{
  private:
     string bill_Id;
     string bill_issuedDate;
     Customer * cus;

  public:
     Bill();
     Bill(string bId,string bIssuedDate,Customer * cst);
     ~Bill();
};


//Report class


class Report{
  private:
     string customer_Details;
     string sCart_Details;
     string order_details;
     string item_Details;
     shopping_cart *rep;
     Item * itml[5];



  public:
     Report();
     Report(string cDetails,string sDetails,string oDetails,string
iDetails,shopping_cart *repl);
     ~Report();
     void list_PreviousOrders();
     void list_sCart_Items();
     void list_restock_Items(int z, Item * itml1);
     void list_detailsCostomer();


};




//Customer class function decleration



Customer::Customer( string cname, string caddress, string cID)
{
    customerName = cname;
    address = caddress;
    customer_ID = cID;
    cout << "Run overload constructor of Customer class" << endl;
}
void Customer::assignShoppingCart(shopping_cart * sc)
{
  shoppingc = sc;
}

void Customer::MakeOrder(int i,Order * ord)
{
  order[i] = ord;
}

void Customer::assignBill(int x,Bill * bl)
{
  bill[x] = bl;
}

void Customer::setDeliveryDetails()
{

}


void Customer::displayCustomer()
{
}


void Customer::setPaymentDetails()
{

}


Customer::~Customer()
{
  cout<<"Customer class has been deleted"<<endl;
}



//Order class function decleration



Order:: Order (string corderID,  string date,Customer * c,Payment * p1)
{
     orderID = corderID;
     dateCreated = date;
     cus = c;
     p = p1;
     cout << "Run overload constructor of Order class" << endl;
}

void Order :: orderItem(int v,Item * tyul)
{
  tyu[v] = tyul;
}

void Order::displayOrders()
{

}

Order::~Order()
{
  cout<<"Order class has been deleted"<<endl;
}



//Item class function decleration
```

```
Item::Item(string iname, string iID, float iPrice, string iDes,shopping_cart *
qw,Report *repal)
{
  itemName = iname;
  itemID = iID;
  itemPrice = iPrice;
  itemDescription = iDes;
  qwe = qw;
  repa = repal;
}

void Item::assignOrderItem(int r,Order *orct)
{
  orc[r] = orct;
}
void Item::displayItem()
{

}

void Item::deleteItem()
{

}

Item::~Item()
{
  cout<<"Item class has been deleted"<<endl;
}



//Shopping cart class function decleration



shopping_cart::shopping_cart(string scart_id,int sqty,string
sdateadded,Customer * cust,Report * sml1)
  {
    cart_id = scart_id;
    qty = sqty;
    dateadded = sdateadded;
    cus = cust;
    sml = sml1;
    cout << "Run overload constructor of shopping_cart class" << endl;
  }

void shopping_cart::addCartItem(int y,Item * it)
{
  itm1[y] = it;
}
void shopping_cart::checkout()
{

}

void shopping_cart::viewCart()
{

}

void shopping_cart::update_Quantity()
{

}

shopping_cart::~shopping_cart()
{
  cout<<"shopping_cart class has been deleted"<<endl;
}



//Payment function decleration


Payment::Payment(float pamount,string ppaymentType,int pcardNo,int pcardCVC,
string pcardExp_date,Order *oq1 )
{
  amount = pamount;
  paymentType = ppaymentType;
  cardNo = pcardNo;
  cardCVC = pcardCVC;
  cardExp_Date = pcardExp_date;
  oq = oq1;
  cout << "Run overload constructor of Payment class" << endl;
}


void Payment::validatePayment()
{

}


void Payment::confirmPayment()
{
}

void Payment::displayDetails()
{

}

Payment::~Payment()
{
  cout<<"Payment class has been deleted"<<endl;
}



//Delivery class function decleration



Delivery::Delivery(string dID, float cPaid, string dMethod)
{
  delivery_ID = dID;
  cashPaid = cPaid;
  delivery_Method = dMethod;
}

void Delivery::displayDeliveryStatus()
{

}

Delivery::~Delivery()
{
  cout<<"Delivery class has been deleted"<<endl;
}



//Bill class function decleration



Bill::Bill(string bId,string bIssuedDate,Customer * cst)
{
  bill_Id = bId;
  bill_issuedDate = bIssuedDate;
  cus = cst;
}
```

```
Bill::~Bill()
{
  cout<<"Bill class has been deleted"<<endl;
}



//Report class function decleration


Report::Report(string cDetails,string sDetails,string oDetails,string
iDetails,shopping_cart *repl)
{
  customer_Details = cDetails;
  sCart_Details = sDetails;
  order_details =  oDetails;
  item_Details = iDetails;
  rep = repl;
}


void Report::list_PreviousOrders()
{

}


void Report::list_sCart_Items()
{

}


void Report::list_restock_Items(int z, Item * itml1)
{
  itml[z] = itml1;
}


void Report::list_detailsCostomer()
{

}


Report::~Report()
```

```
{
  cout<<"Report class has been deleted"<<endl;
}




int main()
{
  //customer class
  Customer *c1;
  c1 = new Customer("s", "ghsfhsi", "1234");

   Order * o1;

  Payment *p1;
  p1 = new Payment(123,"VISA",34543,234, "22/10/2020", o1 );


  o1 = new Order("12345", "2000/10/20",c1,p1);

  delete c1;
  delete o1;
  delete p1;
}
```

Code without Relationships……

```
#include <iostream>
#include <string>
using namespace std;



//CLASSES



//CUSTOMER CLASS


class customer{


  private:
    int customer_ID;
    string customerName;
    string address;
    string phone;
    string deliveryDetails;
    string paymentDetails;


  public:
    customer();
    ~customer();
    customer(int customer_ID,string customerName,string address,int phone,string deliveryDetails,string paymentDetails);


    //Accessors
    void setCustomer(int id,string name,string address,string phNo,string delDetails,string payDetails);


    //Mutators
    int getCustomer_ID();
    string getPhone();
    string getCustomerName();
    string getAddress();
    string getDeliveryDetails();
    string getPaymentDetails();
```

```
};



//ITEM CLASS



class Item
{
private:
                                        string itemName;
                                        int itemID;
 float itemPrice;
                                        string itemDescription;

public:
                                        Item();
                                        ~Item();
                                        Item(string iname, int iID,
float iPrice, string itemDescription);
```

```
                                void setItem(string
iname,int iID,float iPrice,string iDes);

                                string getItemName();

                                int getItemID();

                                float getItemPrice();

                                string
getItemDescription();


};



//SHOPPING CART CLASS



class shopping_cart

{

  private:

     int cart_id;

     int qty;

     string dateadded;
  public:
     shopping_cart();
     shopping_cart(int scart_id,int sqty,string sdateadded);
     ~shopping_cart();
     void setScart(int scart_id,string sdateadded);
     void setQty(int sqty);
     int getCartID();
     int getQty();
     string getDateAdded();
     void checkout();

};



//ORDER CLASS



class Order
{
 private:
  int order_id;
  string dateCreated;
  int qty;

 public:
  Order ();
  ~Order ();
  void setOrder(int oID, string dCreated);
  int getoID();
  string getdCreated();
  void displayOrderStatus();
};

//PAYMENT CLASS

 class Payment {
  private:
    float amount;
    string paymentType;
```

```
        int cardNo;

        int cardCVC;

        string cardExp_Date;


    public:

        Payment();

        ~Payment();

        Payment(float pamount,string ppaymentType,int
pcardNo,int pcardCVC, string pcardExp_date );


        void setPayment(float pamount,string ppaymentType,int
pcardNo,int pcardCVC, string pcardExp_date);


        float getamount();

        string getpaymentType();

        int getcardNo();

        int getcardCVC();

        string getcardExp_Date();


        void validatePayment();
        void confirmPayment();

        void displayDetails();

  };



//DELIVERY CLASS


class Delivery

{

  private:

    int delivery_ID;

    float cashPaid;

    string delivery_Method;


  public:

  Delivery();

  ~Delivery();

  void generateDeliveryReport(int dID, float cPaid, string
dMethod);

  void setDelivery(int dID,float cPaid, string dMethod);
  int getdID();
  float getcPiad();
  string getdMethod();

};

//BILL CLASS


class Bill{
  private:
    int bill_Id;
    string bill_issuedDate;

  public:
    void setBillID(int bId);
    void setBillIssuedDate(string bIssuedDate);
    int getBillID();
    string getBillIssuedDate();
};
```

```
//REPORT CLASS



//REPORT CLASS


class Report{


  private:
      //customer
      int customer_ID;
      string customerName;
      string address;
      string phone;
      string deliveryDetails;
      string paymentDetails;
  //item
  string itemName;
                                 int itemID;
  float itemPrice;
                                 string itemDescription;

  //shopping cart
  int cart_id;
  int qty;
  string dateadded;

  //order
  int order_id;
  string dateCreated;


public:

 Report();
 ~Report();
```

```
    Report(int customer_ID,string customerName,string
address,int phone,string deliveryDetails,string
paymentDetails);


    Report(string iname, int iID, float iPrice, string
itemDescription);


    Report(int scart_id,int sqty,string sdateadded);






    //Accessors
    void setCustomer(int id,string name,string address,string
phNo,string delDetails,string payDetails);


    void setItem(string iname,int iID,float iPrice,string iDes);
                                        string getItemName();


    void setScart(int scart_id,string sdateadded);
    void setQty(int sqty);


    void setOrder(int oID, string dCreated);
    void displayOrderStatus();



    //Mutators
    int getCustomer_ID();
    string getPhone();
    string getCustomerName();
    string getAddress();
    string getDeliveryDetails();
    string getPaymentDetails();


    int getItemID();
                                        float getItemPrice();
    string getItemDescription();


    int getCartID();
    int getQty();
```

```
    string getDateAdded();

    void checkout();


    int getoID();

    string getdCreated();


};






//FUNCTION DECLERATION

//CUSTOMER CLASS FUNCTIONS





customer::customer(){


  customer_ID = 0;

  customerName = "";
  address = "";
  phone = "";
  deliveryDetails = "";
  paymentDetails = "";
}


customer::~customer()
{
}


void customer::setCustomer(int id,string name,string
add,string phNo,string delDetails,string payDetails){
  customer_ID = id;
  customerName = name;
  address = add;
  phone = phNo;
  deliveryDetails = delDetails;
  paymentDetails = payDetails;
}



int customer::getCustomer_ID()
{
  return customer_ID;
}



string customer::getPhone(){
  return phone;
}



string customer::getCustomerName(){

  return customerName;
}
```

```
string customer::getAddress(){

  return address;
}


string customer::getDeliveryDetails(){

  return deliveryDetails;
}


string customer::getPaymentDetails(){

  return paymentDetails;
}
```

```
//ITEM CLASS FUNCTIONS


Item::Item()

{

                                        itemName = "";

                                        itemID = 0;

                                        itemPrice = 0;

                                        itemDescription = "";

}


Item::~Item()

{

}


Item::Item(string iname, int iID, float iPrice, string itemDescription)

{

}
```

```
void Item::setItem(string iname, int iID, float iPrice, string iDes)

{

                                        itemName = iname;

                                        itemID = iID;

                                        itemPrice = iPrice;

                                        itemDescription = iDes;

}


string Item::getItemName()

{

                                        return itemName;

}


int Item::getItemID()

{

                                        return itemID;

}


float Item::getItemPrice()
```

```
{
                                return itemPrice;
}

string Item::getItemDescription()
{
                                return itemDescription;
}



//SHOPPING CART CLASS FUNCTIONS




  shopping_cart::shopping_cart()
 {
   cart_id = 0;
   dateadded = "";
```

```
}

shopping_cart::~shopping_cart()
{
}

void shopping_cart::setScart(int scart_id,string sdateadded)
{
  cart_id = scart_id;
  dateadded = sdateadded;

}

void shopping_cart::setQty(int sqty)
{
  qty = sqty;
}

int shopping_cart::getCartID()
{
  return cart_id;
 }

 int shopping_cart::getQty()
 {
  return qty;
 }

 string shopping_cart::getDateAdded()
 {
  return dateadded;
 }



 void shopping_cart::checkout()
  {
  }

Order::Order(){
 int oID = 0;
```

```
  string dCreated = "";
  int iqty = 0;
}



//ORDER CLASS FUNCTIONS



Order::~Order()
{
}

void Order::setOrder(int oID, string dCreated)
{
  order_id = oID;
  dateCreated = dCreated;
}
```

```
int Order::getoID()
{
  return order_id;
}

string Order::getdCreated()
{
  return dateCreated;
}

//PAYMENT CLASS FUNCTION

Payment::Payment()
{
  amount=0;
  paymentType="";
  cardNo=0;
  cardCVC=0;
  cardExp_Date="";

}

  Payment::~Payment()
  {

  }

  Payment::Payment(float pamount,string ppaymentType,int
pcardNo,int pcardCVC, string pcardExp_date)
  {

  }

  void Payment::setPayment(float pamount, std::string
ppaymentType, int pcardNo, int pcardCVC, std::string
pcardExp_date)
  {
    amount = pamount;
```

```
}

  paymentType = ppaymentType;

  cardNo = pcardNo;

  cardCVC = pcardCVC;

  cardExp_Date = pcardExp_date;


}


float Payment::getamount()

{

  return amount;

}

string Payment::getpaymentType()

{

  return paymentType;

}

int Payment::getcardNo()

{

  return cardNo;

}

int Payment::getcardCVC()
```

```
 {
   return cardCVC;
 }
 string Payment::getcardExp_Date()
 {
   return cardExp_Date;
 }

//Delivery Class Functions


Delivery::Delivery()
{
   int delivery_ID = 0;
   float cashPaid = 0;
   string delivery_Method = "";
}

Delivery::~Delivery()
{
}
```

```
void Delivery::generateDeliveryReport(int dID, float cPaid, string dMethod)

{

  delivery_ID = dID;

  cashPaid = cPaid;

  delivery_Method = dMethod;

}


void Delivery::setDelivery(int dID, float cPaid, std::string dMethod)

{

  delivery_ID = dID;

  cashPaid = cPaid;

  delivery_Method = dMethod;

}


int Delivery::getdID()

{

  return delivery_ID;
```

```
}

float Delivery::getcPiad()
{
  return cashPaid;
}

string Delivery::getdMethod()
{
  return delivery_Method;
}




//BILL CLASS FUNCTIONS


void Bill::setBillID(int bId)
{
  bill_Id=bId;
}
void Bill::setBillIssuedDate(string bIssuedDate)
{
  bill_issuedDate=bIssuedDate;
}
int Bill::getBillID()
{
  return bill_Id;
}
string Bill::getBillIssuedDate()
{
  return bill_issuedDate;
}


 //Report


//REPORT CLASS FUNCTIONS
```

```
Report::Report(){

customer_ID = 0;
customerName = "";
address = "";
phone = "";
deliveryDetails = "";
paymentDetails = "";

itemName = "";
                                itemID = 0;
                                itemPrice = 0;
                                itemDescription = "";

cart_id = 0;
dateadded = "";

int oID = 0;
string dCreated = "";
int iqty = 0;
}



Report::~Report()
{
}

void Report::setCustomer(int id,string name,string add,string phNo,string delDetails,string payDetails){
  customer_ID = id;
  customerName = name;
  address = add;
  phone = phNo;
  deliveryDetails = delDetails;
  paymentDetails = payDetails;
}

Report::Report(string iname, int iID, float iPrice, string itemDescription)
```

```
{

}

void Report::setItem(string iname, int iID, float iPrice, string
iDes)

{

                                                            itemName = iname;

                                                            itemID = iID;

                                                            itemPrice = iPrice;

                                                            itemDescription = iDes;

}


void Report::setScart(int scart_id,string sdateadded)

  {

    cart_id = scart_id;

    dateadded = sdateadded;


  }


void Report::setOrder(int oID, string dCreated)

{
  order_id = oID;

  dateCreated = dCreated;

}




int Report::getCustomer_ID()

{

  return customer_ID;

}



string Report::getPhone(){

  return phone;

}



string Report::getCustomerName(){


  return customerName;
```

```
}


string Report::getAddress(){

  return address;
}


string Report::getDeliveryDetails(){

  return deliveryDetails;
}


string Report::getPaymentDetails(){

  return paymentDetails;
}
```

```
}

string Report::getItemName()

{

                                                                                                        return itemName;

}


int Report::getItemID()

{

                                                                                                        return itemID;

}


float Report::getItemPrice()

{

                                                                                                        return itemPrice;

}


string Report::getItemDescription()

{

                                                                                                        return itemDescription;

}
```

```
void Report::setQty(int sqty)
 {
   qty = sqty;
 }

 int Report::getCartID()
 {
   return cart_id;
 }

 int Report::getQty()
 {
   return qty;
 }

 string Report::getDateAdded()
 {
   return dateadded;
 }
```

```
  void Report::checkout()
   {
   }



int Report::getoID()
{
  return order_id;
}

string Report::getdCreated()
{
  return dateCreated;
}
```

```
//MAIN PROGRAM



int main(){
  customer c1,c2;
  Item i1,i2,i3,i4,i5;
  shopping_cart s,s1,s2,s3;
  Order O1,O2;
  Payment p1;
  Bill b1;
  Report r,r1,r3,r4,r5,r6,R;



  c1.setCustomer(1234,"Dimalka Heshan", "kelapara
padikkama", "0760012641", "bus", "VISA");
```

```
  c2.setCustomer(3456,"Nimna Thiranjaya", "kelapara" ,
"0761234567", "car", "COD");

  i1.setItem("Fresh Milk", 1234, 500.00, "good");
  i2.setItem("Coconut Oil", 1235, 345.00, "good");
  i3.setItem("Mazarella Cheese", 1236, 275.00, "good");
  i4.setItem("Organic Potato", 1237, 80.00, "good");
  i5.setItem("Baby Soap", 1238, 220.00, "good");

  s.setScart(1234,"2020/10/17");
  s1.setQty(5);
  s2.setQty(2);
  s3.setQty(10);

  O1.setOrder(11243, "11/10/2020 10:10AM");
  O2.setOrder(11244, "12/7/2020 7:30AM");

  p1.setPayment(5250.00,"Credit
card",1004545435,400,"21/21");
```

```
 b1.setBillID(2450);
 b1.setBillIssuedDate("11/10/2020");



 r.setCustomer(1234,"Dimalka Heshan", "kelapara padikkama", "0760012641", "bus", "VISA");


 r1.setItem("Fresh Milk", 1234, 500.00, "good");



 r3.setScart(1234,"2020/10/17");
 r4.setQty(5);
 r5.setQty(2);
 r6.setQty(10);

 R.setOrder(11243, "11/10/2020 10:10AM");
```

```
//CUSTOMER



  cout<<"----------------------------CUSTOMER DETAILS---------------
-----------"<<endl;
  cout<<"--------------------------------------------------------------------
"<<endl<<endl;



  cout<<"Customer ID: "<< c1.getCustomer_ID()<<endl;
  cout<<"Customer Name: "<< c1.getCustomerName()<<endl;
  cout<<"Customer Address: "<< c1.getAddress()<<endl;
  cout<<"Customer Phone no: "<< c1.getPhone()<<endl;
  cout<<"Customer Delivery: "<<
c1.getDeliveryDetails()<<endl;
  cout<<"Customer Payment: "<<
c1.getPaymentDetails()<<endl;
  cout<<endl;

  cout<<"Customer ID: "<< c2.getCustomer_ID()<<endl;
  cout<<"Customer Name: "<< c2.getCustomerName()<<endl;
  cout<<"Customer Address: "<< c2.getAddress()<<endl;

  cout<<"Customer Phone no: "<< c2.getPhone()<<endl;

  cout<<"Customer Delivery: "<<
c2.getDeliveryDetails()<<endl;

  cout<<"Customer Payment: "<<
c2.getPaymentDetails()<<endl;

  cout<<endl;



//ITEM



  cout<<"---------------------------------ITEM DETAILS------------------
---------"<<endl;

  cout<<"---------------------------------------------------------------------
--"<<endl<<endl;


  cout << "Item Name: " << i1.getItemName() << endl;

  cout << "Item ID: " << i1.getItemID() << endl;
  cout << "Item Price: " << i1.getItemPrice() << endl;
  cout << "Item Description: " << i1.getItemDescription() <<
endl;
  cout<<endl;

  cout << "Item Name: " << i2.getItemName() << endl;
  cout << "Item ID: " << i2.getItemID() << endl;
  cout << "Item Price: " << i2.getItemPrice() << endl;
  cout << "Item Description: " << i2.getItemDescription() <<
endl;
  cout<<endl;

  cout << "Item Name: " << i3.getItemName() << endl;
  cout << "Item ID: " << i3.getItemID() << endl;
  cout << "Item Price: " << i3.getItemPrice() << endl;
  cout << "Item Description: " << i3.getItemDescription() <<
endl;
  cout<<endl;

  cout << "Item Name: " << i4.getItemName() << endl;
  cout << "Item ID: " << i4.getItemID() << endl;
  cout << "Item Price: " << i4.getItemPrice() << endl;
  cout << "Item Description: " << i4.getItemDescription() <<
endl;
  cout<<endl;

  cout << "Item Name: " << i5.getItemName() << endl;
  cout << "Item ID: " << i5.getItemID() << endl;
  cout << "Item Price: " << i5.getItemPrice() << endl;
  cout << "Item Description: " << i5.getItemDescription() <<
endl;
  cout<<endl;
  cout<<endl;
  cout<<endl;




//SHOPPING CART
```

```
  cout<<"----------------------------SHOPPING CART--------------------
----------"<<endl;

  cout<<"---------------------------------------------------------------------
--"<<endl;

  cout << "Shopping Cart ID :
"<<s.getCartID()<<"\t\t\t\t"<<"Created on
"<<s.getDateAdded()<<endl;

  cout<<endl;

  cout<<endl;

  cout<<"Item ID"<<"\t\t"<<"Item
Name"<<"\t\t\t\t\t\t"<<"Quantity"<<endl;

  cout<<"_"<<endl;


cout<<i1.getItemID()<<"\t\t"<<i1.getItemName()<<"\t\t\t\t\t
\t\t"<<s1.getQty()<<endl;


cout<<i2.getItemID()<<"\t\t"<<i2.getItemName()<<"\t\t\t\t\t
\t\t"<<s2.getQty()<<endl;


cout<<i3.getItemID()<<"\t\t"<<i3.getItemName()<<"\t\t\t\t\t
"<<s3.getQty()<<endl;

  cout<<endl;

  cout<<endl;
```

```
//ORDER



  cout<<"---------------------------------ORDER DETAILS----------------
----------"<<endl;
  cout<<"----------------------------------------------------------------------
--"<<endl<<endl;
  cout<<"Customer ID: "<< c1.getCustomer_ID()<<endl;
  cout<<"Customer Name: "<< c1.getCustomerName()<<endl;
  cout<<"Customer Address: "<< c1.getAddress()<<endl;
  cout<<"Customer Phone no: "<< c1.getPhone()<<endl;
  cout<<endl;

  cout<<"Order ID : "<<O1.getoID()<<"\t\t\t\t"<<"Order
Created on "<<O1.getdCreated()<<endl;
  cout<<endl;
  cout<<"Code"<<"\t\t"<<"Item Name"<<"\t\t\t\t"<<"Price"<<"\t\t\t"<<"Qty"<<"\t\t"<<"Tot. Price"<<endl;



cout<<i1.getItemID()<<"\t\t"<<i1.getItemName()<<"\t\t\t\t"<<i1.getItemPrice()<<"\t\t\t\t"<<s1.getQty()<<"\t\t\t"<<i1.getItemPrice()*s1.getQty()<<endl;



cout<<i3.getItemID()<<"\t\t"<<i3.getItemName()<<"\t\t"<<i3.getItemPrice()<<"\t\t\t\t"<<s3.getQty()<<"\t\t\t"<<i3.getItemPrice()*s3.getQty()<<endl;

  cout<<endl;

  cout<<endl;



  cout<<"\t\t\t\t\t\t\t\t\t\t\t\t\t"<<"SUB TOTAL : "<<i1.getItemPrice()*s1.getQty()+i3.getItemPrice()*s3.getQty()<<endl;

  cout<<endl;

  cout<<endl;
```

```
//PAYMENT


  cout<<"------------------------------PAYMENT DETAILS---------------
-----------"<<endl;

  cout<<"---------------------------------------------------------------------
--"<<endl<<endl;

  cout<<"Amount :"<<p1.getamount()<<endl;

  cout<<"Payment Type :"<<p1.getpaymentType()<<endl;

  cout<<"Card No :"<<p1.getcardNo()<<endl;

  cout<<"Card CVC :"<<p1.getcardCVC()<<endl;

  cout<<"Card Expire Date :"<<p1.getcardExp_Date()<<endl;


  cout<<endl;

  cout<<endl;




//Delivery
```

```
  cout<<"----------------------------DELIVERY DETAILS---------------
-----------"<<endl;

  cout<<"-----------------------------------------------------------------
--"<<endl<<endl;

  cout<<endl;

Delivery d1;


d1.setDelivery(1000, 1050.00, "Card Payment");

cout<<"Customer ID: "<< c1.getCustomer_ID()<<endl;

cout<<"Order ID : "<<O1.getoID()<<endl;

cout<<"Bill ID : "<<d1.getdID()<<endl;

cout<<endl;

cout<<"Total Bill :
"<<d1.getcPiad()<<".00"<<"\t\t\t\t"<<"Payment Method :
"<<d1.getdMethod()<<endl;

cout<<endl;


cout<<"\t\t\t\t\t*Payment Has Been Made*"<<endl;

cout<<endl;

cout<<"Delivery Has Been Made On 11/10/2020 At
1:15PM"<<endl;
```

```
cout<<endl;

cout<<endl;


//BILL


  cout<<"-------------------------BILL------------------------
---"<<endl;

  cout<<"-----------------------------------------------------
--"<<endl<<endl;

  cout<<"\t\t\t\t\t\t\t\tGrocery mania "<<endl;

  cout<<"\t\t\t\t\t\t\tCall us +9476002641 "<<endl;

  cout<<"\t\t\t\t\t\tEmail us support@grocerymania.com
"<<endl<<endl;

  cout<<"\t\t"<<"Qty"<<"\t\t\t"<<"Item
Name"<<"\t\t\t"<<"Price"<<"\t\t\t"<<"Tot.Price"<<endl;


cout<<"\t\t"<<s1.getQty()<<"\t\t\t"<<i1.getItemName()<<"\t
\t\t"<<i1.getItemPrice()<<"\t\t\t\t\t"<<i1.getItemPrice()*s1.g
etQty()<<endl;


cout<<"\t\t"<<s3.getQty()<<"\t\t\t"<<i3.getItemName()<<"\t
```

```
"<<i3.getItemPrice()<<"\t\t\t\t\t"<<i3.getItemPrice()*s3.getQ
ty()<<endl<<endl;


 cout<<"\t\t\t\t\t\t\t\t\t\t\t\t"<<"SUB TOTAL :
"<<i1.getItemPrice()*s1.getQty()+i3.getItemPrice()*s3.getQty
()<<endl;


 cout<<"\t\t\t\t\t\t\t\t"<<"THANK YOU!"<<endl<<endl;


cout<<"Bill ID-"<<b1.getBillID()<<"\t\t\t\t\t\t\t\t"<<"Bill
issued date-"<<b1.getBillIssuedDate()<<endl;


cout<<"------------------------------------------------------------------------
"<<endl;




//REPORT


cout<<"----------------------------REPORT DETAILS-----------------------
----"<<endl;
  cout<<"----------------------------------------------------------------------
"<<endl<<endl;
```

```
  cout<<"Customer ID: "<< r.getCustomer_ID()<<endl;
  cout<<"Customer Name: "<< r.getCustomerName()<<endl;
  cout<<"Customer Address: "<< r.getAddress()<<endl;
  cout<<"Customer Phone no: "<< r.getPhone()<<endl;
  cout<<"Customer Delivery: "<< r.getDeliveryDetails()<<endl;
  cout<<"Customer Payment: "<<
r.getPaymentDetails()<<endl;
  cout<<endl;

  cout << "Item Name: " << r1.getItemName() << endl;
  cout << "Item ID: " << r1.getItemID() << endl;
  cout << "Item Price: " << r1.getItemPrice() << endl;
  cout << "Item Description: " << r1.getItemDescription() <<
endl;
  cout<<endl;

  cout << "Shopping Cart ID :
"<<r3.getCartID()<<"\t\t\t\t"<<"Created on
"<<r3.getDateAdded()<<endl;
  cout<<endl;
  cout<<endl;
  cout<<"Item ID"<<"\t\t"<<"Item
Name"<<"\t\t\t\t\t\t"<<"Quantity"<<endl;
  cout<<"_"<<endl;

cout<<r4.getItemID()<<"\t\t"<<r4.getItemName()<<"\t\t\t\t\
t\t\t"<<r4.getQty()<<endl;


  cout<<endl;
  cout<<endl;


  cout<<"Customer ID: "<< r.getCustomer_ID()<<endl;
  cout<<"Customer Name: "<< r.getCustomerName()<<endl;
  cout<<"Customer Address: "<< r.getAddress()<<endl;
  cout<<"Customer Phone no: "<< r.getPhone()<<endl;
  cout<<endl;


  cout<<"Order ID : "<<R.getoID()<<"\t\t\t\t"<<"Order
Created on "<<R.getdCreated()<<endl;
  cout<<endl;
  cout<<"Code"<<"\t\t"<<"Item
Name"<<"\t\t\t\t"<<"Price"<<"\t\t\t"<<"Qty"<<"\t\t"<<"Tot.
Price"<<endl;



cout<<r1.getItemID()<<"\t\t"<<r1.getItemName()<<"\t\t\t\t"
<<r1.getItemPrice()<<"\t\t\t\t"<<r4.getQty()<<"\t\t\t"<<r1.g
etItemPrice()*r4.getQty()<<endl;



  cout<<"\t\t\t\t\t\t\t\t\t\t\t\t\t"<<"SUB TOTAL :
"<<r1.getItemPrice()*r4.getQty()<<endl;

  cout<<endl;

  cout<<endl;





/*


  cout<<"--------------------------------------------------"<<endl;
  cout<<"Customer ID: "<< c2.getCustomer_ID()<<endl;
  cout<<"Customer Name: "<< c2.getCustomerName()<<endl;
  cout<<"Customer Address: "<< c2.getAddress()<<endl;
  cout<<"Customer Phone no: "<< c2.getPhone()<<endl;
  cout<<"Customer Delivery: "<<
c2.getDeliveryDetails()<<endl;
  cout<<"Customer Payment: "<<
c2.getPaymentDetails()<<endl;
  cout<<endl;

  cout<<"Order ID : "<<O2.getoID()<<"\t\t\t\t"<<"Order
Created on "<<O2.getdCreated()<<endl;
  cout<<endl;
  cout<<"Code"<<"\t\t"<<"Item
Name"<<"\t\t\t\t"<<"Price"<<"\t\t"<<"Qty"<<endl;

cout<<i2.getItemID()<<"\t\t"<<i2.getItemName()<<"\t\t\t\t"
<<i2.getItemPrice()<<"\t\t\t"<<s2.getQty()<<endl;

  cout<<"--------------------------------------------------"<<endl;
  cout<<endl; */


  return 0;
}
```